

**Skúška z predmetu 2-INF-132 Úvod do distribuovaných algoritmov**  
(13. 12. 2011)

1 Zmysel pre orientáciu grafu  $G$  sa definuje nasledovne: Nahradíme každú hranu  $G$  dvoma protismernými orientovanými hranami a každú hranu nového grafu označíme  $O(\log \Delta)$  bitovou značkou, kde  $\Delta$  je maximálny stupeň vrchola. Takéto značkovanie je “zmysel pre orientáciu” v  $G$ , ak existuje funkcia, ktorá pre dve cesty vychádzajúce z jedného vrchola určí iba pomocou postupnosti značiek na hranách, či končia v tom istom bode.

Nájdite aspoň dva neizomorfné zmysly pre orientáciu v hyperkocke.

2 Uvažujme anonymný kruh (t.j. všetky procesory sú identické). Povieme, že algoritmus  $A$  ráta funkciu  $f : \{0, 1\}^* \mapsto \{0, 1\}$  podľa nasledovného. Pre vstup  $X = x_1, \dots, x_n$  zoberieme  $n$  procesorov zoradených do kruhu, pričom  $i$ -ta kópia (v cyklickom číslovaní) má vstupnú hodnotu  $x_i$  a vo všetkých beží algoritmus  $A$ . Nakoniec všetky kópie zistia  $f(X)$  a skončia. Funkcia je cyklická, t.j.  $f(x_1, x_2, \dots, x_n) = f(x_2, x_3, \dots, x_n, x_1)$ . Dokážte, že ak nejaký algoritmus  $A$  ráta  $f$ , potom  $f$  je konštantná (t.j. nezávisí od  $X$ ).

3 Uvažujme nasledovný algoritmus na voľbu šéfa v úplných grafoch (je identický s algoritmom z prednášky s tým rozdielom, že zajaté vrcholy sa pri ďalšom pokuse o zajatie nepýtajú rodiča, ale rozhodnú sa na základe svojho  $[level, id]$ ). Ukážte, že tento algoritmus vyžaduje  $\Omega(n^2)$  správ (t.j. nájdite konštantu  $c$  a pre každé dosť veľké  $n$  nájdite úplný graf s  $n' \geq n$  vrcholmi, rozdelenie identifikátorov a časovanie správ tak, aby sa vymenilo aspoň  $cn'^2$  správ).

```
const:  N      : integer
        ID     : integer
        Neigh  : [1..N-1] link
var:    leader : boolean
        state  : {active, captured, killed}
        level  : integer
        parent : link
        i      : integer
```

Init:

```
state := active, level := 0, leader := false
```

Code:

```
for i = 1 to N - 1 do
    send <capture, [level, ID]> to Neigh[i]
    receive <accept> from Neigh[i]
    level ++
leader := true
for i = 1 to N - 1 do
    send <leader, ID> to Neigh[i]
```

Dead:

**loop forever**

On receipt <capture, [level<sub>i</sub>, id<sub>i</sub>]> from Neigh[i]:

```
if state ∈ {active, killed} and [leveli, idi] > [level, ID]
    state := captured
    parent := Neigh[i]
    send <accept> to parent
    goto Dead
else if state = captured
    if [leveli, idi] > [level, ID]
        send <accept> to Neigh[i]
        parent := Neigh[i]
```

On receipt <leader, id<sub>i</sub>> from Neigh[i]:

*Skonči algoritmus*

4 Uvažujme asynchrónny kruh, v ktorom je jedna chybná linka. Akákoľvek správa, poslaná po chybnéj linke je úspešne odoslaná, ale nikdy nie je doručená. Predpokladajme, že v kruhu je zvolený šéf, ktorý pozná veľkosť kruhu. Nájdite algoritmus, ktorý identifikuje chybnú linku s použitím  $O(n \log n)$  správ.